

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Факультет компьютерных систем и информационных технологий
Кафедра информатики и программной инженерии

УТВЕРЖДАЮ

Декан факультета компьютерных
систем и информационных технологий

Кочевский А. А.

« 19 »

2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

«Платформа .NET»

09.03.03 Прикладная информатика

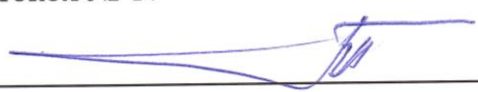
«Прикладная информатика в экономике»

Разработчик:

доцент  Письменский А.В.

ФОС рассмотрен и одобрен на заседании кафедры информатики и
программной инженерии
от 18 апреля 2023 г., протокол № 17

Заведующий кафедрой

 Кочевский А.А.

Луганск 2023 г.

**Паспорт
фонда оценочных средств по учебной дисциплине
«Платформа .NET»**

**Перечень компетенций (элементов компетенций),
формируемых в результате освоения учебной дисциплины**

№ п/п	Код контроли руемой компетен ции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины	Этапы формирования (семестр изучения)
1	ПК-2.1	Способен использовать современные языки и системы программирования, технологии проектирования программного обеспечения	Тема 1. Тема 2.	начальный (5)
2	ПК-2.2	Способен сформулировать требования к разрабатываемому программному обеспечению, выполнить его реализацию и оформить техническую документацию на его компоненты	Тема 3. Тема 4.	начальный (5)
3	ПК-2.3	Способен осуществлять проектирование программного обеспечения конкретной ИС и разработку технической документации на ее компоненты	Тема 5. Тема 6. Тема 7.	начальный (5)

**Показатели и критерии оценивания компетенций,
описание шкал оценивания**

№ п/п	Код контролируемо й компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1.	ПК-2.1	Знать: Общие сведения о платформе .NET. Платформа .NET Framework. Достоинства .NET. Среда CLR. Модель выполнения кода в CLR. Структура программы. Типы и переменные. Операторы. Уметь: использовать классы и объекты в языке C#; свойства объектов; закрытые, защищенные и открытые поля; основные парадигмы программирования; методы класса; платформу программирования .NET Framework; языки C#, SQL, XML	Тема 1. Основные концепции платформы .NET Тема 2. Особенности языка C#	Вопросы для обсуждения (в виде докладов и сообщений), лабораторные работы, контрольные работы,
2.	ПК-2.2	Знать: Классы. Участники. Параметры типа. Базовые классы. Поля. Методы. Параметры. Тело метода и локальные переменные. Статические методы и методы экземпляра. Переопределение метода. Переопределение свойств. Ключевое слово base. Запрет переопределения методов. Абстрактные классы. Абстрактные члены классов. Абстрактные методы. Абстрактные свойства. Отказ от реализации абстрактных членов. Пример абстрактного класса. Перегрузка методов. Уметь: Создавать и использовать виртуальные методы и абстрактные классы	Тема 3. Классы и объекты Тема 4. Виртуальные методы и абстрактные классы	Вопросы для обсуждения (в виде докладов и сообщений), лабораторные работы, контрольные работы,
3.	ПК-2.3	Знать: Функции-члены. Конструкторы. Конструкторы экземпляров.	Тема 5. Другие функции-члены Тема 6. Делегаты,	Вопросы для обсуждения (в виде докладов и

		Инициализаторы конструктора. Инициализаторы переменных экземпляров. Выполнение конструктора. Конструкторы по умолчанию. Закрытые конструкторы. Необязательные параметры конструктора экземпляра. Статические конструкторы. Свойства. Индексаторы. События. Операторы. Методы завершения. Делегаты. Уметь: Применение делегатов. Добавление и удаление методов в делегате. Анонимные методы. Лямбды. События. Введение в интерфейсы. Применение интерфейсов. Явная реализация интерфейсов. Модификаторы доступа. Реализация интерфейсов в базовых и производных классах. Наследование интерфейсов. Интерфейсы в обобщениях. Интерфейсные ссылки. Ключевое слово <code>as</code>. Ключевое слово <code>is</code>. Интерфейсные свойства и индексаторы. Копирование объектов. Интерфейс <code>ICloneable</code>.	события и лямбды Тема 7. Интерфейсы	сообщений), лабораторные работы, контрольные работы,
--	--	---	--	---

Фонды оценочных средств по дисциплине «Платформа .NET»

Вопросы для обсуждения (в виде докладов и сообщений):

1. Исторический обзор развития компонентного программирования: Smalltalk, CORBA, COM, Java.
2. Предпосылки и история возникновения платформы .NET и языка C#.

3. Классификация основных типов платформы .NET.
4. Вспомогательные средства C#.
5. Система запросов, интегрированная в язык C#.
6. SQL-запросы в языке C#.
7. Взаимодействие COM и .NET компонент.
8. Сравнительный анализ языков Java и C#.
9. Библиотеки базовых классов.NET.
- 10.Безопасность платформы .NET.
- 11.Средства параллелизма платформы .NET.
- 12.Принципы асинхронного ввода/вывода в ОС Windows.
- 13.Базовые функции асинхронного ввода/вывода платформы .NET.
- 14.Последовательный асинхронный код на языке C#.
- 15.Архитектура Джеффри Рихтера (алгоритм Async Enumerator).
- 16.Языковые средства асинхронности в C# 5.0.

Критерии и шкала оценивания по оценочному средству доклад, сообщение

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Доклад (сообщение) представлен(о) на высоком уровне (студент в полном объеме осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, владеет профильным понятийным (категориальным) аппаратом и т.п.)
4	Доклад (сообщение) представлен(о) на среднем уровне (студент в целом осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, допустив некоторые неточности и т.п.)
3	Доклад (сообщение) представлен(о) на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)
2	Доклад (сообщение) представлен(о) на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)

Лабораторные работы

ВВЕДЕНИЕ

Цели лабораторных работ:

- формирование практических навыков организации и использования при решении задач динамических структур данных;
- изучение наиболее распространенных алгоритмов решения задач с использованием сложных структур данных.

Содержание работ одинаково для всех лабораторных работ.

Язык программирования – язык Си/C++.

Порядок выполнения работы

1. Изучить необходимый материал по теме лабораторной работы, руководствуясь методическими указаниями к теме.
2. Получить допуск к работе, ответив на контрольные вопросы.
3. Разработать структуру данных для представления основных исходных данных и/или результатов для выданного преподавателем варианта задания.
4. Разработать алгоритм и написать программу решения задачи.
5. Подготовить набор тестов и отладить программу.
6. Оформить отчет по лабораторной работе. Отчет по работе включает следующие пункты:
 - титульный лист;
 - описание полученного задания;
 - анализ задачи;
 - структура основных входных и выходных данных;
 - алгоритм решения задачи;
 - текст программы;
 - набор тестов;
 - результаты отладки и их анализ;
 - скриншоты, демонстрирующие работу программы;
 - выводы.
7. Защитить лабораторную работу. Защита состоит в обсуждении выбранной структуры данных и алгоритма решения задачи, ответе на контрольные вопросы, решении контрольных задач.

Описание лабораторной установки

Лабораторные работы выполняются на ПЭВМ типа IBM, работающей под управлением ОС Windows XP/7/10.

Лабораторная работа 1

Тема: События. Указание действия в случае события. Подключение метода обработчика событий к событию.

Цель работы: познакомиться с понятием "событие" и их обработчиками.

Лабораторная работа 2

Тема: Приложения Windows Forms.

Цель работы: освоить работу с приложениями Windows Forms.

Лабораторная работа 3

Тема: Изучение элементов Button, ListBox, MessageBox, PictureBox, CheckBox, RadioButton, RichTextBox, ToolBarButton.

Цель работы: овладение практическими навыками использования элементов ListBox, MessageBox, PictureBox, ToolBarButton, RadioButton, RichTextBox, CheckBox, Button для разработки пользовательского интерфейса.

Лабораторная работа 4

Тема: Пространство имен Drawing.

Цель работы: познакомиться с пространством имен Drawing.

Лабораторная работа 5

Тема: Конфигурирование стандартных компонентов и создание пользовательского интерфейса.

Цель работы: изучить конфигурирование стандартных компонентов и научиться создавать пользовательский интерфейс.

Лабораторная работа 6

Тема: Оконные приложения. Управление внешним видом формы.

Цель работы: освоить создание оконных приложений и управление внешним видом формы.

Лабораторная работа 7

Тема: Изучение классов и типов пространства имен SYSTEM.IO. Организация работы с каталогами и файлами.

Цель работы: изучить классы и типы пространства имен SYSTEM.IO, освоить работу с каталогами и файлами.

Лабораторная работа 8

Тема: Протоколирование событий.

Цель работы: изучить протоколирование событий.

Лабораторная работа 9

Тема: Разработка приложений для работы с изображениями и файлами параметров.

Цель работы: изучить методы работы с изображениями и файлами параметров.

Лабораторная работа 10

Тема: Серверы COM. Разработка приложений для взаимодействия с Open-OfficeWriter.

Цель работы: освоить разработку приложений для взаимодействия с Open-OfficeWriter.

Лабораторная работа 11

Тема: Базы данных и XML.

Цель работы: освоить работу с базами данных и языком разметки XML.

Критерии и шкала оценивания по оценочному средству лабораторная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Лабораторная работа выполнена на высоком уровне (правильность выполнения 90-100%)
4	Лабораторная работа выполнена на среднем уровне (правильность выполнения 75-89%)
3	Лабораторная работа выполнена на низком уровне (правильность выполнения 50-74%)
2	Лабораторная работа выполнена на неудовлетворительном уровне (правильность выполнения менее чем на 50%)

Вопросы к контрольным работам:

Контрольные задания.

Задание 1.

Разработать тип, позволяющий вычислять корень n -й степени из числа методом Ньютона с заданной точностью.

Предусмотреть возможность сравнения полученного результата со значением, рассчитываемым с помощью метода Math.Pow библиотеки классов .NET Framework.

В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

Задание 2.

Разработать тип, позволяющий конвертировать неотрицательное десятичное значение целого числа в строку, содержащую двоичное представление этого значения.

В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

Задание 3.

Разработать тип, реализующий алгоритм Евклида для вычисления НОД двух целых чисел (http://en.wikipedia.org/wiki/Euclidean_algorithm). Метод должен принимать выходной параметр, содержащий значение времени, необходимое для выполнения расчета. Добавить к разработанному типу дополнительную функциональность в виде перегруженных методов вычисления НОД для трех и т.д. целых чисел.

Добавить к разработанному типу метод, реализующий алгоритм Стейна (бинарный алгоритм Эвклида) для расчета НОД двух целых чисел (http://en.wikipedia.org/wiki/Binary_GCD_algorithm). Метод должен принимать выходной параметр, содержащий значение времени, необходимое для выполнения расчетов. Добавить к разработанному типу дополнительную функциональность в виде перегруженных методов вычисления НОД для трех и т.д. целых чисел.

Разработать метод, позволяющий сравнивать результаты работы разработанных методов на одинаковых наборах данных. (Разработать метод для отображения результатов работы с использованием гистограммы. Метод должен принимать в качестве двух обязательных параметров время выполнения, и в качестве необязательных параметров – ориентацию гистограммы (вертикальная или горизонтальная) и цвет для отображения ее полос).

Создать unit-тесты для тестирования разработанных методов.

В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

Задание 4.

Разработать класс «вектор» для работы с трехмерными векторами.

Перегрузить для класса операции, допустимые для работы с трехмерными векторами.

Создать unit-тесты для тестирования разработанных методов.

В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

Задание 5.

Разработать класс «многочлен» для работы с многочленами от одной переменной.

Перегрузить для класса операции, допустимые для работы с многочленами.

Создать unit-тесты для тестирования разработанных методов.

В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

Задание 6.

Разработать тип для работы с матрицами.

Реализовать методы, позволяющие выполнять основные операции над матрицами, предусмотрев возможность их выполнения, в противном случае должно генерироваться исключение.

В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

Создать unit-тесты для тестирования методов разработанного типа.

Задание 7.

Создать класс Автомобиль с полями марка, мощность двигателя, цвет, максимальная скорость. Необходимо определить полный конструктор, а также функцию поиска наиболее скоростного автомобиля из двух, и вывода информации на экран. Предусмотреть возможность проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

Задание 8.

Создать класс Еда с полями состав, количество килокалорий, масса. Необходимо определить полный конструктор, а также функцию поиска калорийного блюда из двух, и вывода информации на экран. Предусмотреть возможность проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

Задание 9.

Создать класс Самолет с полями количество пассажиров, крейсерская скорость, тип самолета (для дальних или ближних перелетов), название авиакомпании. Необходимо определить полный конструктор, а также функцию поиска наиболее вместительного самолета из двух, и вывода информации на экран. Предусмотреть возможность проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

Задание 10.

Создать класс Стол с полями: высота стола, ширина, толщина стола, материал, из которого изготовлен стол. Необходимо определить полный конструктор, а также функцию определения площади поверхности стола и вывода информации на экран. Предусмотреть возможность проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

Задание 11.

Создать класс Смартфон с полями: масса, цвет, оперативная память, емкость аккумулятора. Необходимо определить полный конструктор, а также функцию расчета производительности Смартфона (Оперативная память * емкость аккумулятора) и вывода информации на экран. Предусмотреть возможность проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

Критерии и шкала оценивания по оценочному средству контрольная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

ЗАДАНИЕ **по дисциплине** **«Платформа .NET»**

Подготовить реферат, согласно выбранной теме.

Требования к оформлению

Реферат должен соответствовать теме и содержать 15 - 20 страниц машинописного текста. Титульный лист оформляется по стандарту, далее следует план изложения с указанием страниц в тексте. В тексте устанавливаются поля: слева – 20 мм, справа – 15 мм, сверху – 20 мм, снизу – 20 мм, выравнивание по ширине. Шрифт текста – Times New Roman, размер шрифта – 12, начертание – обычный, интервал – полуторный. Для заголовков установить размер шрифта – 14, начертание – полужирный, выравнивание по центру. Абзац начинается с красной строки с отступом 1,25 см, нумерация страниц начинается с 3 номера, внизу, по центру, арабскими цифрами. По всему тексту вставить все необходимые рисунки, таблицы, графики. В конце указать список используемой литературы или ссылки на интернет-источники. По материалам реферата подготовить статью в научный журнал, тезисы для участия в конференции. Представить реферат в электронном и бумажном вариантах.

Таблица выбора тем

№	Содержание
1.	Сравнительный анализ платформ программирования .Net и J2EE
2.	Язык разметки графических интерфейсов XAML
3.	Автоматизация сборки программного продукта в среде Visual Studio: msbuild-сценарии и командные bat-файлы
4.	Классы и объекты в языке C#
5.	Свойства объектов
6.	Добавление полей в класс
7.	Элементы и атрибуты в XML
8.	Алгоритм Рейнгольда-Тилфорда визуализации бинарных деревьев
9.	Сравнение пакетов Java и пространств имён C#
10.	Компонентное программирование: Smalltalk, CORBA, COM, Java
11.	Итераторы и генераторы итераторов (оператор yield)
12.	Лямбда-выражения
13.	Технология LINQ to SQL
14.	Взаимодействие COM и .NET компонент на примере работы с Excel из C#
15.	Параллелизм в приложениях с графическим интерфейсом пользователя
16.	Базовые функции асинхронного ввода/вывода платформы .NET

Критерии и шкала оценивания по оценочному средству реферат

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Реферат представлен на высоком уровне (студент в полном объеме осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, владеет профильным понятийным (категориальным) аппаратом и т.п.). Оформлен в соответствии с требованиями, предъявляемыми к данному виду работ.
4	Реферат представлен на среднем уровне (студент в целом осветил рассматриваемую проблематику, привел аргументы в пользу своих суждений, допустив некоторые неточности и т.п.). В оформлении допущены некоторые неточности в соответствии с требованиями, предъявляемыми к данному виду работ.
3	Реферат представлен на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.). В оформлении допущены ошибки в соответствии с требованиями, предъявляемыми к данному виду работ.
2	Реферат представлен на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)

Тесты:

1. Какая из перечисленных WinAPI функций работает с UNICODE:

- а) CreateFileW;
- б) CreateFileA;
- в) Ни одна из перечисленных.

2. Особенностью модального окна является:

- а) Модальные окна не могут быть созданы пользовательскими приложениями;
- б) Невозможно работать с родительскими окнами, пока модальное окно не будет закрыто;
- в) Для создания модального окна необходимо обязательно использовать ресурсы в исполняемом модуле.

3. Кусты реестра (hives):

- а) Всегда формируются динамически;
- б) Использовались только в старых версиях Windows (3.1);
- в) Загружаются исключительно из файлов;
- г) Загружаются из файла или формируются динамически.

4. За авторизацию локальных пользователей отвечает процесс:

- а) System Idle Process;
- б) explorer.exe;
- в) LSASS.

5. Что такое ENUM?

- а) Используется для инициализации переменных;
- б) Используется для определения констант;
- в) Он используется для определения переменных;
- г) Нет верного ответа.

6. Что делает метод Dispose?

- а) Закрыть соединение;
- б) Временно избавиться от соединения;
- в) Удаляет его из памяти;
- г) Все вышеперечисленное.

7. Особенность локальной переменной

- а) Она должна быть объявлена в методе;
- б) представляет объект класса;
- в) Может использоваться в любом месте программы;
- г) Она должна использоваться в классе.

8. В чем разница между Convert.ToInt32 и Int.Parse?

- а) Оба одинаковы;

- б) Convert.ToInt32 не может обработать нулевые значения, он выдаст ошибку ArgumentNullException;
- в) Int.Parse не может обработать нулевые значения, он выдает ошибку ArgumentNullException;
- г) Оба могут обрабатывать нулевые значения.

9. C# не поддерживает:

- а) абстракция;
- б) полиморфизм;
- в) множественное наследование;
- г) наследство.

10. Какое из следующих утверждений верно в отношении управляемого кода?

- а) Управляемый код, который выполняется поверх Windows;
- б) Управляемый код CLR;
- в) Управляемый код – это то место, где код является сборщиком мусора;
- г) Управляемый код для компиляторов.

11. Можем ли мы переопределить частные виртуальные методы?

- а) Да;
- б) Нет.

12. Какие из следующих сборок можно сохранить в Global Assembly Cache?

- а) private;
- б) shared;
- в) protected;
- г) friend.

13. Сборка – это:

- а) декларативный синтаксис Java;
- б) свойства C#;
- в) события C#;
- г) Коллекция файлов, которые представляются программисту в виде одной DLL или EXE.

14. Какие из следующих утверждений правильно определяют .NET Framework?

- а) Это среда для разработки, создания, развертывания и выполнения настольных приложений, веб-приложений и веб-служб;
- б) Это среда для разработки, создания, развертывания и выполнения распределенных приложений;
- в) Это среда для разработки и исполнения приложений Windows;

г) Это среда для разработки, создания, развертывания и выполнения только веб-приложений.

15. Какое из следующих утверждений ИСТИННО в отношении .NET CLR?

1. Он обеспечивает нейтральную для языка среду разработки и исполнения.
 2. Это гарантирует, что он не авторизован для доступа.
 3. Предоставляет сервисы для запуска «управляемых» приложений.
 4. Ресурсы мусора.
 5. Предоставляет сервисы для запуска «неуправляемых» приложений.
- а) только 1 и 2;
б) только 1, 2 и 4;
в) 1, 2, 3, 4;
г) только 4 и 5.

16. Что можно использовать в считывателе данных до чтения?

- а) `getvalue`;
б) `getString`;
в) `getnumber`;
г) `None`.

17. Может ли интерфейс быть создан непосредственно?

- а) Да;
б) Нет.

18. Какие из перечисленных ниже являются действительными счетчиками производительности .NET CLR JIT?

1. Общая память, используемая для компиляции JIT.
 2. Средняя память, используемая для компиляции JIT.
 3. Количество методов, которые не удалось скомпилировать со стандартным JIT.
 4. Процент JIT-компиляции процессорного времени.
 5. Процент памяти, выделенной для компиляции JIT.
- а) 1, 2;
б) 1, 5;
в) 3, 4;
г) 4, 5.

19. Автоматическая страничная подкачка возможна в:

- а) `datareader`;
б) `dataset`;
в) `datatable`;
г) Все варианты.

20. Какой класс унаследован от класса?

- а) Sealed;
- б) Gather;
- в) Static;
- г) Constru.

21. Особенности переменных только для чтения:

- а) Декларация и инициализация разделены;
- б) Распределяется во время компиляции;
- в) Распределяется во время выполнения;
- г) Все вышеперечисленное.

22. Событие имеет _____ в качестве типа возврата по умолчанию:

- а) Нет типа возврата для событий;
- б) String;
- в) Double;
- г) Integer.

23. Код, предназначенный для общезыковой среды выполнения, известен как:

- а) Наследие;
- б) Управляемый код;
- в) Неуправляемый;
- г) Родной код.

24. Какой метод вызывается для объекта SqlDataAdapter для загрузки вашего набора данных с данными?

- а) Обновить;
- б) Заполнить;
- в) Собрать;
- г) Загрузить.

25. Могут ли частные переменные уровня класса наследоваться?

- а) Да, и мы можем получить к ним доступ;
- б) Нет, и мы не можем получить к ним доступ;
- в) Да, но мы не можем получить к ним доступ;
- г) Все вышеперечисленное неверно.

26. Есть ли ошибки в этом конструкторе EmployeeMgmt: Public int EmployeeMgmt {emp_id = 100; }

- а) Тип возврата;
- б) Формальные параметры;
- в) ошибок нет;
- г) Имя.

27. Что верно в отношении интерфейса и абстрактных методов?

- а) Мы можем написать только один абстрактный метод внутри интерфейса;
- б) Ни один метод не является абстрактным внутри интерфейса;
- в) В абстрактном методе;
- г) Ничего из вышеперечисленного.

28. Какой объект вы можете использовать?

- а) Объект сеанса;
- б) Объект сервера;
- в) Объект ответа;
- г) Объект приложения.

29. Два метода с одинаковым именем, но с разными параметрами:

- а) Перегрузка;
- б) Мультиплексирование;
- в) Дуплекс;
- г) Загрузка.

30. В чем разница между методом `Convert.ToString (str)` и `str.ToString ()`?

- а) Функция `Convert.ToString (str)` обрабатывает `NULL`, а функция `str.ToString ()` – нет. Будет выброшено исключение `NULL`;
- б) Функция `str.ToString ()` обрабатывает `NULL`, а `Convert.ToString (str)` – нет. Будет выброшено исключение `NULL`;
- в) Оба могут обрабатывать `NULL`;
- г) Никто не может обращаться с `NULL`.

31. Какое из этих определений строк предотвратит обратную косую черту в C#?

- а) строка `s = #.n` Тестовая строка .;
- б) строка `s = @.n` Тестовая строка .;
- в) строка `s = ..n` тестовая строка .;
- г) строка `s = .n` тестовая строка .;

32. Разница между `Convert.ToString ()` и `ToString ()`:

- а) `Convert.ToString ()` обрабатывает нулевые значения, но `ToString ()` нет;
- б) `Convert.ToString ()` обрабатывает только нулевые значения;
- в) `ToString ()` обрабатывает нулевые значения, но `Convert.ToString ()` нет;
- г) Вывод `ToString ()` в соответствии с предоставленным форматом.

33. Как можно отсортировать элементы массива в порядке убывания?

- а) вызывая метод `Sortdescending()`;
- б) вызывая метод `Sort()`;
- в) Вызывая методы `Sort()` и затем `Reverse()`;
- г) вызывая метод `Sortreverse()`.

34. Что из следующего составляет .NET Framework?

1. Приложения ASP.NET.
 2. CLR.
 3. Framework Class Library.
 4. Приложения WinForm.
 5. Службы Windows.
- а) 2, 5;
б) 2, 1;
в) 2, 3;
г) 3, 4.

35. Нужна структура для проведения опросов (порядка 25-ти вопросов с ответами true-false). Требования к структуре:

Проинициализировать каждый ответ в значение true.

Минимизировать объем памяти для каждого опроса.

Какое хранилище лучше всего использовать?

- а) `BitVector32 answers = new BitVector32(1);`
б) `BitVector32 answers = new BitVector32(-1);`
в) `BitArray answers = new BitArray(1);`
г) `BitArray answers = new BitArray(-1);`

36. Из каких частей состоит NET Runtime?

- а) Common Language Runtime;
б) Набор базовых классов;
в) Виртуальная машина C#;
г) Сборщик мусора.

37. Создается класс для сравнения специально отформатированных строк (без использования collation по умолчанию) с реализацией интерфейса `Comparable<string>`.

Какой фрагмент кода следует использовать?

- а)

```
public class Person : Comparable<string> {  
    public int CompareTo(string other) {  
        ...  
    }  
}
```
- б)

```
public class Person : Comparable<string> {  
    public bool CompareTo(string other) {  
        ...  
    }  
}
```
- в)

```
public class Person : Comparable<string> {  
    public bool CompareTo(object other) {  
        ...  
    }  
}
```

```
}
```

```
г) public class Person : IComparable<string> {  
    public int CompareTo(object other) {  
        ...  
    }  
}
```

38. Требуется создать динамическую сборку (dynamic assembly) MyAssembly и сохранить ее на диске.

Какой фрагмент кода следует использовать?

- а) `AssemblyName myAssemblyName = new AssemblyName();
myAssemblyName.Name = "MyAssembly";
AssemblyBuilder myAssemblyBuider = AppDomain.CurrentDomain.DefineDynamicAssembly(myAssemblyName, AssemblyBuiderAccess.Run);
myAssemblyBuilder.Save("MyAssembly.dll");`
- б) `AssemblyName myAssemblyName = new AssemblyName();
AssemblyBuilder myAssemblyBuider = AppDomain.CurrentDomain.DefineDynamicAssembly(myAssemblyName, AssemblyBuiderAccess.RunAndSave);
myAssemblyBuilder.Save("MyAssembly.dll");`
- в) `AssemblyName myAssemblyName = new AssemblyName("MyAssembly");
AssemblyBuilder myAssemblyBuider = AppDomain.CurrentDomain.DefineDynamicAssembly(myAssemblyName, AssemblyBuiderAccess.Save);
myAssemblyBuider.Save("c:\\MyAssembly.dll");`
- г) `AssemblyName myAssemblyName = new AssemblyName();
myAssemblyName.Name = "MyAssembly";
AssemblyBuilder myAssemblyBuider = AppDomain.CurrentDomain.DefineDynamicAssembly(myAssemblyName, AssemblyBuiderAccess.Save);
myAssemblyBuilder.Save("MyAssembly.dll");`

39. Дан код:

```
public delegate void SimpleDelegate(string str);  
public class A  
{  
    public void simple(object o)  
    {  
        Console.WriteLine("A: " + str); //1
```

```

    }
}

public class B : A
{
    public void simple(string str)
    {
        Console.WriteLine("B: " + o); //2
    }
}

```

Какой из методов будет вызван в результате такого кода:

```

B b = new B( );
SimpleDelegate act = new SimpleDelegate(b.simple);
act("HI!");

```

- a) 1;
- б) 2;
- в) Ошибка компиляции.

40. Требуется написать фрагмент кода, который:

1. Получает названия каждого приостановленного сервиса.
2. Передает название в метод Add объекта Collection1.

Какой фрагмент кода следует использовать?

- a)

```

ManagementObjectSearcher searcher =
    new ManagementObjectSearcher(
        "Select * from Win32_Service", "State = 'Paused' ");
foreach (ManagementObject svc in searcher.Get( )) {
    Collection1.Add(svc["DisplayName"]);
}

```
- б)

```

ManagementObjectSearcher searcher =
    new ManagementObjectSearcher(
        "Select * from Win32_Service where State = 'Paused' ");
foreach (ManagementObject svc in searcher.Get( )) {
    Collection1.Add(svc["DisplayName"]);
}

```
- в)

```

ManagementObjectSearcher searcher =
    new ManagementObjectSearcher( );
searcher.Scope = new ManagementScope("Win32_Service ");
foreach (ManagementObject svc in searcher.Get( )) {
    if ((string) svc["State"] == "Paused") {
        Collection1.Add(svc["DisplayName"]);
    }
}

```
- г)

```

ManagementObjectSearcher searcher =
    new ManagementObjectSearcher(

```

```

        "Select * from Win32_Service");
foreach (ManagementObject svc in searcher.Get( )) {
    if ((string) svc["State"] == " 'Paused' ") {
        Collection1.Add(svc["DisplayName"]);
    }
}

```

41. Создается приложение, которое получает значения из некоторой секции конфигурационного XML-файла.

Пример секции:

```

<ProjectSection name="Project1">
    <role name="administrator" />
    <role name="manager" />
    <role name="support" />
</ProjectSection>

```

Необходимо описать класс Role и убедиться, что он инициализирован значениями, полученными из файла конфигурации.

Какой сегмент кода следует использовать?

- a) `public class Role : ConfigurationElement {`
 `internal string _ElementName = "role";`
 `[ConfigurationProperty("name", RequiredValue = true)]`
 `public string Name {`
 `get {`
 `return ((string)base[("name")]);`
 `}`
 `}`
`}`
- б) `public class Role : ConfigurationElement {`
 `internal string _ElementName = "name";`
 `private string _name;`
 `[ConfigurationProperty("role", RequiredValue = true)]`
 `public string Name {`
 `get {`
 `return _name;`
 `}`
 `}`
`}`
- в) `public class Role : ConfigurationElement {`
 `internal string _ElementName = "name";`
 `[ConfigurationProperty("role")]`
 `public string Name {`
 `get {`
 `return ((string)base[("role")]);`
 `}`
 `}`
`}`

```

    }
r) public class Role : ConfigurationElement {
    internal string _ElementName = "role";
    private string _name;
    [ConfigurationProperty("name")]
    public string Name {
        get {
            return _name;
        }
    }
}
}

```

42. SMTP сервер доступен в локальной подсети и называется smtp.contoso.com. Адрес источника me@contoso.com, адрес получателя you@contoso.com.

Какой фрагмент кода позволит отправить e-mail?

- a) `MailAddress addrFrom = new MailAddress("me@contoso.com", "Me");`
`MailAddress addrTo = new MailAddress("you@contoso.com", "You");`
`MailMessage message = new MailMessage(addrFrom, addrTo);`
`message.Subject = "Greetings!";`
`message.Body = "Test";`
`message.Dispose( );`
- б) `MailAddress addrFrom = new MailAddress("me@contoso.com");`
`MailAddress addrTo = new MailAddress("you@contoso.com");`
`MailMessage message = new MailMessage(addrFrom, addrTo);`
`message.Subject = "Greetings!";`
`message.Body = "Test";`
`SmtpClient client = new SmtpClient("smtp.contoso.com");`
`client.Send(message);`
- в) `MailAddress addrFrom = new MailAddress("me@contoso.com", "Me");`
`MailAddress addrTo = new MailAddress("you@contoso.com", "You");`
`MailMessage message = new MailMessage(addrFrom, addrTo);`
`message.Subject = "Greetings!";`
`message.Body = "Test";`
`SocketInformation info = new SocketInformation( );`
`Socket client = new Socket(info);`
`System.Text.ASCIIEncoding enc = new System.Text.ASCIIEncoding( );`
`byte[] msgBytes = enc.GetBytes(message.ToString( ));`
`client.Send(msgBytes);`
- г) `string strSmtpClient = "smtp.contoso.com";`
`string strFrom = "me@contoso.com";`
`string strTo = "you@contoso.com";`
`string strSubject = "Greetings!";`
`string strBody = "Test";`
`MailMessage msg =`

```
new MailMessage(strFrom, strTo, strSubject, strSmtpClient);
```

43. Требуется вызвать функцию из Вашего кода, используя platform invoke. Что следует сделать?

- а) Создать класс для функций в DLL, а после создать прототип методов в Вашем коде;
- б) Зарегистрировать свою сборку, а затем связать ее с COM;
- в) Экспортировать библиотеку типов для Вашего кода;
- г) Импортировать библиотеку типов как сборку и после создать экземпляр COM объекта.

44. Требуется создать метод в классе-контейнере. Нужно убедиться, что создаваемый в классе метод возвратит тип, который может работать в foreach блоке.

Какому критерию должен удовлетворять метод?

- а) Метод должен возвращать IEnumerator или IEnumerable;
- б) Метод должен возвращать итератор;
- в) Метод должен принимать коллекцию в качестве параметра;
- г) Метод должен возвращать IComparable.

45. Как описать класс Age так, чтобы коллекция Age объектов могла быть отсортирована?

- а)

```
public class Age {  
    public int Value;  
    public object CompareTo(object obj) {  
        if (obj is Age) {  
            Age _age = (Age) obj;  
            return Value.CompareTo(obj);  
        }  
        throw new ArgumentException("object not an Age");  
    }  
}
```
- б)

```
public class Age {  
    public int Value;  
    public object CompareTo(int iValue) {  
        try {  
            return Value.CompareTo(iValue);  
        } catch {  
            throw new ArgumentException("object not an Age");  
        }  
    }  
}
```
- в)

```
public class Age : IComparable {  
    public int Value;
```

```

public int CompareTo(object obj) {
    if (obj is Age) {
        Age _age = (Age) obj;
        return Value.CompareTo(_age.Value);
    }
    throw new ArgumentException("object not an Age");
}
}

```

```

г) public class Age : IComparable {
    public int Value;
    public int CompareTo(object obj) {
        try {
            return Value.CompareTo(((Age) obj).Value);
        } catch {
            return -1;
        }
    }
}

```

46. Требуется указать тип, который удовлетворяет следующим критериям:

Является целочисленным.

Не превышает значения 65535.

Какой тип следует использовать?

- а) System.IntPtr;
- б) int;
- в) System.UInt8;
- г) System.UInt16.

47. Дан код для вызова функции из Win32 с использованием platform invoke:

```
int rc = MessageBox(hWnd, text, caption, type);
```

Как правильно определить прототип метода:

- а) [DllImport(@"C:\WINDOWS\system32\user32.dll")]


```
public static extern int MessageBox(int hWnd, String text,
                                     String caption, uint type);
```
- б) [DllImport("user32")]


```
public static extern int MessageBoxA(int hWnd, String text,
                                     String caption, uint type);
```
- в) [DllImport("user32")]


```
public static extern int Win32API_User32_MessageBox
      (int hWnd, String text, String caption, uint type);
```
- г) [DllImport("user32")]


```
public static extern int MessageBox(int hWnd, String text,
```

String caption, uint type);

48. Требуется вывести список языковых и региональных кодов. Какой фрагмент кода следует для этого использовать?

- а) `foreach (CultureInfo culture in CultureInfo.GetCultures(CultureTypes.ReplacementCultures)) {
 // Вывод информации...
}`
- б) `foreach (CultureInfo culture in CultureInfo.GetCultures(CultureTypes.NeutralCultures)) {
 // Вывод информации...
}`
- в) `CultureInfo culture = new CultureInfo("");
CultureTypes types = culture.CultureTypes;
// Вывод информации...`
- г) `foreach (CultureInfo culture in CultureInfo.GetCultures(CultureTypes.SpecificCultures)) {
 // Вывод информации...
}`

49. Дан фрагмент класса Employee для COM:

```
public class Employee {  
    string employeeId, employeeName, jobTitleName;  
    public string GetName( ) { return employeeName; }  
    public string GetTitle( ) { return jobTitleName; }  
    ...  
}
```

COM интерфейс должен также обеспечить совместимость с будущими версиями класса Employee.

Какой метод для генерации COM интерфейса необходимо выбрать?

а) Определить интерфейс для класса и добавить следующий атрибут к определению класса:

```
[ClassInterface(ClassInterfaceType.None)]  
public class Employee : IEmployee {
```

б) Добавить следующий атрибут к определению класса:

```
[ComVisible(true)]  
public class Employee {
```

в) Добавить следующий атрибут к определению класса:

```
[ClassInterface(ClassInterfaceType.AutoDual)]  
public class Employee {
```

г) Добавить следующий атрибут к определению класса:

```
[ClassInterface(ClassInterfaceType.None)]  
public class Employee {
```

50. Разрабатывается метод для шифрования данных с помощью алгоритма Data Encryption Standard (DES).

Метод принимает в качестве параметров массив байт message, который нужно зашифровать, ключ шифрования – key и вектор инициализации – iv. Зашифрованные данные нужно записать в объект MemoryStream.

Какой фрагмент кода следует использовать?

a) DES des = new DESCryptoServiceProvider();
des.BlockSize = message.Length;
ICryptoTransform crypto = des.CreateEncryptor(key, iv);

MemoryStream cipherStream = new MemoryStream();
CryptoStream cryptoStream = new CryptoStream(cipherStream, crypto,
CryptoStreamMode.Write);
cryptoStream.Write(message, 0, message.Length);

б) DES des = new DESCryptoServiceProvider();
ICryptoTransform crypto = des.CreateEncryptor(key, iv);
MemoryStream cipherStream = new MemoryStream();
CryptoStream cryptoStream = new CryptoStream(cipherStream, crypto,
CryptoStreamMode.Write);
cryptoStream.Write(message, 0, message.Length);

в) DES des = new DESCryptoServiceProvider();
ICryptoTransform crypto = des.CreateDecryptor(key, iv);

MemoryStream cipherStream = new MemoryStream();
CryptoStream cryptoStream = new CryptoStream(cipherStream, crypto,
CryptoStreamMode.Write);
cryptoStream.Write(message, 0, message.Length);

г) DES des = new DESCryptoServiceProvider();
ICryptoTransform crypto = des.CreateEncryptor();

MemoryStream cipherStream = new MemoryStream();
CryptoStream cryptoStream = new CryptoStream(cipherStream, crypto,
CryptoStreamMode.Write);
cryptoStream.Write(message, 0, message.Length);

51. Дан следующий код:

```
public delegate void SimpleDelegate(string str);
```

```
public class A
{
    public void simple(string str)
    {
        Console.WriteLine("A: " + str); //1
    }
}
```

```
public class B : A
```

```

{
    public void simple(object o)
    {
        Console.WriteLine("B: " + o); //2
    }
}

```

Какой из методов будет вызван в результате выполнения такого кода:

B b = new B();

SimpleDelegate act = new SimpleDelegate(b.simple);

act("HI!")

а) 1;

б) 2;

в) Ни один из методов вызван не будет;

г) Произойдет ошибка компиляции;

е) Произойдет ошибка времени выполнения из-за невозможности однозначно определить метод для вызова.

52. Необходимо разработать метод, который ищет подстроку в строке для итальянского языка.

Метод принимает строку для поиска – searchList, и строку, которую необходимо найти, – searchValue.

Какой фрагмент кода следует использовать в методе?

а) return searchList.IndexOf(searchValue);

б) CompareInfo comparer = new CultureInfo("it-IT").CompareInfo;
 if (comparer.IndexOf(searchList, searchValue) > 0) {
 return true;
 } else {
 return false;
 }

в) CompareInfo comparer = new CultureInfo("it-IT").CompareInfo;
 return comparer.Compare(searchList, searchValue);

г) CultureInfo comparer = new CultureInfo("it-IT");
 if (searchList.IndexOf(searchValue) > 0) {
 return true;
 } else {
 return false;
 }

53. Изменяются параметры настройки безопасности файла MyData.xml. Требуется сохранить унаследованные правила доступа. Кроме этого нужно обеспечить неизменяемость прав доступа в будущем.

Какой фрагмент кода следует использовать?

а) FileSecurity security = new FileSecurity();
 security.SetAccessRuleProtection(true, true);
 File.SetAccessControl("mydata.xml", security);

- б) FileSecurity security = File.GetAccessControl("mydata.xml");
security.SetAuditRuleProtection(true, true);
File.SetAccessControl("mydata.xml", security);
- в) FileSecurity security = new FileSecurity("mydata.xml",
AccessControlSections.All);
security.SetAccessRuleProtection(true, true);
File.SetAccessControl("mydata.xml", security);
- г) FileSecurity security = File.GetAccessControl("mydata.xml");
security.SetAccessRuleProtection(true, true);

Критерии и шкала оценивания по оценочному средству тесты

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Тесты выполнены на высоком уровне (правильные ответы даны на 90-100% тестов)
4	Тесты выполнены на среднем уровне (правильные ответы даны на 75-89% тестов)
3	Тесты выполнены на низком уровне (правильные ответы даны на 50-74% тестов)
2	Тесты выполнены на неудовлетворительном уровне (правильные ответы даны менее чем на 50% тестов)

Оценочные средства для промежуточной аттестации (экзамен)

1. Свойства CIL и принципы работы.
2. Дуальные интерфейсы (dual-интерфейсы).
3. System. IO. Способы получения информации о дате создания текущего каталога.
4. Создание перегруженных методов.
5. System. IO: создание и сопровождение файлов.
6. System. IO: создание и сопровождение каталогов.
7. Возможности служб Microsoft .NET.
8. Варианты использования класса Command библиотеки ADO.NET.
9. Окружение выполнения кода в .NET.
10. Автоматизация и интерфейс IDispatch.
11. Автоматизация в COM.
12. Диспетчерские интерфейсы (disp-интерфейсы).
13. Интерфейс IUnknown и его методы.
14. Работы со структурой TimeSpan.
15. Последовательность действий при исполнении JIT-compiler.
16. Работа со строковыми переменными: методы IndexOf(), Substring(), свойство Length.
17. Использование блоков using.

18. Переменные в .NET. Принципы объявления, инициализации и использования.
19. Обработка ошибок. Исключительная ситуация типа `FormatException`.
20. Элемент управления `MaskedTextBox`.
21. Принципы работы Microsoft® .NET Access Control Service.
22. Принципы работы Microsoft® .NET Service Bus.
23. Элемент управления `ToolStripContainer`.
24. `Windows Forms` (в платформе .NET).
25. Размерные типы данных `C#`.
26. Ссылочные типы данных `C#`.
27. Исключительные ситуации в платформе .NET.
28. Как называется компилятор, осуществляющий трансляцию IL-кода?
29. Типы серверов. Фабрика классов. Создание COM-объекта.
30. Какая системная утилита используется для внесения и удаления из реестра `Windows` информации о компоненте?
31. Какие их языки можно использовать при написания программ для платформы .NET?
32. Какие методы интерфейса `IUnknown` реализуют механизм подсчета ссылок?
33. Какие специализированные типы данных используются в качестве параметров функций `disp`-интерфейсов в технологии Автоматизации?
34. Каким образом COM-интерфейсы и COM-объекты однозначно идентифицируются в системе?
35. Работа с массивами `C#`.
36. Делегаты и события.
37. Какой из методов интерфейса `IDispatch` предназначен для вызова функций `disp`-интерфейсов?
38. Какой из методов интерфейса `IDispatch` предназначен для получения диспетчерского идентификатора (`DISPID`) функции объекта по ее имени?
39. Синтаксис лямбда-выражений.
40. `System.IO`. Какой метод добавляет строки в файл?
41. Какой элемент управления следует выбрать, когда требуется добавить на форму элемент управления, позволяющий выбирать одно из двух возможных состояний (включено или выключено)?
42. Класс `Control` в .NET.
43. Что происходит при повторном обращении к скомпилированному методу?
44. Компонент платформы .NET Framework CLR (`Common Language Runtime`).
45. Компоненты платформы .NET.

- 46.Метод QueryInterface базового интерфейса IUnknown.
- 47.Определение и использование C# класса.
- 48.Определение статического метода C# класса.
- 49.Оператор return.
- 50.Оператор switch.
- 51.Переопределение виртуальных методов базового класса в классе-потомке на языке C#.
- 52.Основные субъекты технологии Автоматизации COM.
- 53.Варианты хранения COM-компонентов.
- 54.Соглашения имен интерфейсов.
- 55.Использование object.
- 56.Принцип JIT (Just-In-Time) компиляции, используемый для программ .NET.
- 57.Сборка .NET.
- 58.COM интерфейс в своем бинарном представлении представляет собой?...
- 59.Таблица виртуальных методов объекта (виртуальная таблица, vtable объекта).
- 60.Тип данных BSTR.
- 61.Тип данных VARIANT.
- 62.Типы значений C#.
- 63.Пространство имен System.IO.
- 64.Объявление и инициализация двумерного массива на языке C#.
- 65.Объявление и инициализация одномерного массива на языке C#.
- 66.Создания компонентного объекта посредством фабрики классов.
- 67.Создание приложений Windows Forms.
- 68.Через какой параметр метода Invoke передается массив аргументов вызываемой функции disp-интерфейса?
- 69.Что представляет собой выходной формат файла, скомпилированного компилятором C#?
- 70.Что представляет собой делегат?
- 71.Что собой представляет фабрика классов?
- 72.Экземпляры класса Forms (в .NET) представляют собой...
- 73.Элементы управления набора классов .NET Windows Forms (Button, Label, ...) предназначены для...

Критерии и шкала оценивания по оценочному средству промежуточный контроль (экзамен)

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по дисциплине «Платформа .NET» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.03.03 Прикладная информатика.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии факультета компьютерных
систем и информационных
технологий



Ветрова Н. Н.